IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Serial No. 09/740,460
Confirmation No. 4070

SUPPLEMENTAL APPEAL BRIEF

Sir/Madam:

This Supplemental Appeal Brief is filed in response to a Notice of Non-Compliant Brief mailed July 24, 2006. The originally-filed Appeal Brief was filed further to the Notice of Appeal faxed to and received by the U.S. Patent and Trademark Office on November 18, 2004. The Notice of Appeal was filed following mailing of a final Office Action on September 2, 2004. Appellant hereby appeals to the Board of Patent Appeals and Interferences from a final rejection of claims 1-3, 5-15, 17 and 18, and respectfully requests that this appeal be considered by the Board.

I.   REAL PARTY IN INTEREST

The subject application is owned by International Business Machines Corporation, a corporation having its principal place of business at New Orchard Road, Armonk, New York, 10504, as evidenced by the assignment recorded at Reel 011391, Frame 0839.

II.   RELATED APPEALS AND INTERFERENCES

The related application listed below shares a common specification with the application currently on appeal, and is also under appeal at this time.

09/740,531   Appeal Brief filed March 21, 2005

However, because dissimilar art is cited in the present application and the above-mentioned related application, Appellants do not believe that the outcome of this appeal will have any bearing on the Board's decision on the related appeal. No other appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III.   STATUS OF CLAIMS

Claims 1-3, 5-15, 17, and 18 are pending and stand rejected in the captioned case.

IV.   STATUS OF AMENDMENTS

Amendments to the claims have not been filed subsequent to their final rejection. The Appendix hereto therefore reflects the current state of the claims.

V.   SUMMARY OF CLAIMED SUBJECT MATTER

Appellant's claimed invention relates to a software system, computer program product, web server and method for caching a parent server page and a child server page on the web server side. More specifically, Appellant's claimed invention relates to a method for improving cache capacity and efficiency by reducing the amount of cache space required to store "embedded" server pages (i.e., parent server pages containing one or more calls to a child server page).

For example, Java Server Pages (JSP) and servlets may be used to generate dynamic web content. JSPs and servlets can be nested – i.e., one JSP or servlet can call another. A JSP or servlet called by another JSP or servlet is sometimes referred to as "nested" or "embedded." In some cases, the main body of a (parent) JSP may contain one or more calls to other (child) JSPs. When a JSP containing calls to other JSPs is cached in conventional methods, a fully expanded version of each called JSP is embedded at the point in the main code from which it is called. This may result in the storage of multiple copies of the called (child) JSP, since it may be called from more than one place in the main (parent) JSP. In addition to reducing cache capacity, the above approach reduces cache efficiency when child JSPs are cached in fully expanded form within the parent JSP. If, for example, data or code within a child JSP changes in main memory, the cached version of the child JSP must be invalidated. However, since the child JSP is cached in expanded form within the parent JSP, the parent JSP must also be invalidated. Therefore, instead of merely updating the child JSP, the conventional approach reduces cache efficiency and diminishes system overhead by requiring that all cached content (including the parent JSP) be updated. *See* Specification: page 7, lines 3-18; page 29, lines 6-10.

The invention as recited in independent claims 1, 9, 17 and 18 improves cache performance by providing a software system, computer program product, web server and method for caching parent and child server pages on the web server side. More specifically, the invention as recited in claims 1, 9, 17 and 18 improves cache capacity by reducing the amount of cache space required to store "embedded" server pages. The invention also improves cache efficiency by reducing the amount of cached content that has to be updated when a child server page is invalidated.

In some embodiments, the presently claimed software system (claim 1) may include a parent server page (e.g., JSP1 200, Fig. 5), containing a call to a child server page (e.g., JSP2), and a cache (154, Fig. 4) located within a web server (14, Fig. 1) and containing code for both the parent and child server pages. In preferred embodiments of the invention, however, the code for the parent server page does <u>not</u> contain all of the code for the child server page. Instead, the software system may include a link (*see,* e.g., the link within JSP1 206, Fig. 5), which is associated with the call to the child server page, and encapsulating information for locating and

executing the code for the child server page. This allows the parent and child server pages to be cached separately, since the link may be used to locate the child server page in the cache, or to execute the child server page without having to re-execute the parent server page. In this manner, use of the presently claimed link reduces the number of cached lines (i.e., improves cache capacity) by not requiring that the parent server page contain all of the code for the child server page (as in the conventional manner, where the child server page is embedded in expanded form within the parent server page when cached). In addition, use of the presently claimed link improves cache efficiency by not requiring that the parent server page be updated when changes are made to the child server page, since the link preserves the connection to the new, updated version of the child. *See* Specification: page 13, lines 4-18; page 27, line 32 to page 29, line 22.

In other embodiments, the presently claimed computer program product (claim 17), web server (claim 18) and method (claim 9) for caching a parent and a child server page may include: storing code for the parent server page (e.g., JSP1 200, Fig. 5) in a cache (154, Fig. 4) located internal to a web server (14, Fig. 1), such that the code for the parent server page does not contain all lines of code for the child server page. Unlike the conventional method, however, only one copy of the code for the child server page may be stored in the cache (instead of redundant, multiple copies of code for each call to the child server page). For example, a link to the singular copy of the child server page code (*see,* e.g., the link within JSP1 206, Fig. 5) may be cached along with the code for the parent server page; the link may later be used for locating and executing the code for the child server page. In some cases, the link may be associated with more than one call to the child server page. Even though the singular copy of the code for the child server page may be executed (from the cache) more than once, use of the presently claimed link improves cache performance by: (i) reducing the amount of cache space required to store "embedded" server pages, and (ii) reducing the amount of cached content that has to be updated when a child server page is invalidated. *See* Specification: page 13, lines 4-18; page 27, line 32 to page 29, line 22.

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1.   Claims 1-3, 5-15, 17, and 18 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,598,048 to Carneal et al. (hereinafter "Carneal") in view of U.S. Patent No. 6,643,652 to Helgeson et al. (hereinafter "Helgeson").

VII.   ARGUMENT

The contentions of the Appellant with respect to the ground of rejection presented for review, and the basis thereof, with citations of the statutes, regulations, authorities, and parts of the record relied upon are presented herein for consideration by the Board. Details as to why the rejections cannot be sustained are set forth below.

Patentability of claims 1-3, 5-15, 17 and 18 under 35 U.S.C § 103(a)

Claims 1-3, 5-15, 17, and 18 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Carneal in view of Helgeson. To establish a case of *prima facie* obviousness of a claimed invention, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. Second, there must be a reasonable expectation of success. As stated in MPEP 2143.01, the fact that references can be hypothetically combined or modified is not sufficient to establish a *prima facie* case of obviousness. *See In re Mills*, 916 F.2d 680 (Fed. Cir. 1990). Finally, the prior art references must teach or suggest <u>all</u> the claim limitations. *In re Royka*, 490 F.2d 981 (CCPA 1974); MPEP 2143.03 (emphasis added). Specifically, "all words in a claim must be considered when judging the patentability of that claim against the prior art." *In re Wilson* 424 F.2d 1382 (CCPA 1970). Using these standards, Applicants contend that the cited art fails to provide teaching, suggestion or motivation for all limitations of the present claims, and furthermore, cannot be combined or modified to do so. Even if the cited art were combined (without sufficient motivation to do so), the hypothetical combination would still fail to teach or suggest all features of the currently pending claims. Several distinctive features of the present invention are set forth in more detail below.

1. **Carneal fails to provide teaching or suggestion for a software system including a cache located within a web server and containing code for a parent server page <u>and</u> a child server page.**

Independent claim 1 recites in part, "[a] software system supporting distributed web applications, comprising … a cache within a web server, containing code for the parent server page and child server page." Independent claims 9, 17, and 18 recite similar limitations. In this manner, the Specification and present claims disclose a web content caching system where parent and child server pages are each cached on the web server side.

Statements in the final Office Action suggest that Carneal discloses "a cache within a proxy server (e.g., col. 7, lines 63-65), containing code for the parent server page (e.g., col. 8, lines 7-10) and child server page (e.g., col. 8, lines 7-10, an inline object is child page) …" (Final Office Action, page 3). However, and as described in more detail below, Carneal fails to provide teaching or suggestion for: (i) caching parent server pages <u>and</u> child server pages, or (ii) caching server pages within a <u>web server</u>.

Carneal discloses a system and method for reducing the time delay associated with requesting inline objects over a satellite link. As shown in FIG. 6, for example, the system of Carneal comprises a proxy server including: (i) an access point component (70), which runs on the client (browser) side of the satellite link (62) and communicates with web browsers (20), and (ii) a satellite gateway component (72), which runs on the internet (web server) side of the satellite link and communicates with web servers (26). *See*, e.g., Carneal, column 3, lines 8-18, and column 7, line 47 to column 8, line 6. The system of Carneal operates as follows.

In response to a user request, the web server (26) forwards a requested parent file to the satellite gateway (72), which in turn, forwards the parent file over the satellite link (62) to the access point (70) and requesting web browser (20). *See*, e.g., Carneal, column 8, lines 7-29. "When a web server returns a parent file of a web page that has been requested by the user, the satellite gateway component parses the parent file to identify any references to inline objects, and prefetches these objects from the web server." (Carneal, column 3, lines 18-22). The satellite gateway then "forwards the prefetched objects over the satellite link to the access point

component, which in turn, caches the inline objects until requested by the browser." (Carneal, column 3, lines 26-29). In this manner, the system of Carneal reduces the time delay associated with requesting inline objects and reduces traffic over a satellite link by prefetching the inline objects (on the web server side) and caching the prefetched inline objects (on the client side) without waiting for the web browser to receive the parent file and generate requests for the inline objects. *See*, e.g., Carneal, column 3, lines 23-25 and 29-35.

The Examiner admits that "Carneal fails to disclose a cache within a web server" (Final Office Action, page 3). The Appellants agree. As noted above, Carneal only provides teaching for caching inline objects (the alleged "child server page") within a cache (71) located on the client side. Carneal provides absolutely no teaching, suggestion or motivation for storing the inline objects within a cache located within a web server, as presently claimed. As described in more detail below (*see*, Argument 2), Carneal lacks the necessary motivation that would enable one skilled in the art to modify or combine the teachings of Carneal, so that the inline objects were, instead, cached on the web server side. In addition, arguments are provided below to explain why Carneal cannot be modified for caching both parent and child server pages, regardless of where the cache is located.

Contrary to the above Office Action statements, Carneal fails to provide teaching or suggestion for a cache that contains code for both parent and child server pages, regardless of where the cache is located. In the passage cited by the Examiner (col. 8, lines 7-10), Carneal merely states that a parent file may contain references to an inline object (the alleged "child server page"). Though Carneal discloses that the inline objects may be cached (e.g., within object cache 71 of access point 70, as shown in Fig. 6), Carneal does not teach or suggest that the parent file may also be stored along with the inline objects in the same cache. Instead, Carneal specifically states that "after the [parent] file traverses the wireless link, the access point 70 forwards the parent file to the web browser 20" (*See*, Carneal, column 8, lines 22-28; Fig. 6). As such, although Carneal discloses that inline objects may be cached (*See*, Carneal, column 8, lines 29-43), Carneal states that the parent file is forwarded to the requesting web browser and provides absolutely no indication that the parent file may also be cached.

2. **The teachings of Carneal cannot be modified or combined with those of Helgeson to provide the claimed software system for caching server pages in a web server.**

As noted in the above Argument, the Examiner admits that "Carneal fails to disclose a cache within a web server" (Final Office Action, page 3). However, the Examiner suggests that the placement of the cache is a matter of design preference, and therefore, suggests that the teachings of Carneal could be modified to include a cache located in a web server (Final Office Action, pages 3-4). For example, the Examiner suggests that since "Helgeson … disclose[s] a cache within a web server (col. 63, lines 65-67)," it would have been obvious to modify the teachings of Carneal with the web server side cache of Helgeson to provide control over caching in a web content server (Final Office Action, page 4). The Appellants disagree. As set forth below in more detail below, Carneal lacks the necessary teaching, suggestion or motivation that would enable one skilled in the art to make the proposed modification. Therefore, such modification cannot be made.

First of all, Carneal provides no teaching, suggestion or even desirability for caching server pages in a cache located within a web server. The mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680, 16 USPQ2d 1430 (Fed. Cir. 1990); MPEP 2143.01. Appellants assert that the teachings of Carneal cannot be combined or modified with the web server side cache of Helgeson, since Carneal fails to even suggest a desirability for doing so. The mere mention of a cache within a web server (as allegedly taught by Helgeson) does not provide sufficient motivation that would enable one skilled in the art to make the proposed combination or modification.

Second, if the proposed modification were made (without sufficient motivation to do so), the proposed modification would render the invention of Carneal unsatisfactory for its intended purpose. If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984).

As noted above, the intended purpose of Carneal is to reduce a time delay associated with the retrieval of inline objects over a satellite link. *See*, Carneal, Abstract. To accomplish this purpose, Carneal utilizes a proxy server (68, Fig. 6) including an access point component (70) and a satellite gateway component (72) connected via a satellite link (62). The access point is located on a client (browser) side, while the satellite gateway is located on a web server side of the satellite link. Once the satellite gateway receives a requested parent file, the satellite gateway functions to parse the parent file, extract the external reference to the inline object, and forward a request over the Internet to prefetch the inline object. The satellite gateway forwards the prefetched object to the access point, which stores the inline object for later reference in object cache 71 (*See*, Carneal, col. 8, lines 29-42). By placing the object cache within the access point (i.e., on the client-side), inline objects can be prefetched from the Internet and, when needed, can be promptly received by web browser 20, thus, "reducing the effect of the round-trip delay associated with transversal of the satellite link." (Carneal, col. 7, line 66 to col. 8, line 1).

If the teachings of Carneal were modified for caching the prefetched inline objects within a web server (as suggested by the Examiner), instead of client-side access point 70, the system of Carneal would not be able to reduce the time delay associated with the retrieval of inline objects over satellite link 62. Since the modifications proposed by the Examiner would render the invention of Carneal unsatisfactory for its intended purpose, Carneal cannot be modified to provide teaching or suggestion for all limitations of present claims 1, 9, 17, and 18.

Helgeson is merely relied upon to show that caching can be performed on the web server side. Helgeson provides no other teachings relevant to the presently claimed case. In addition, Appellants have proved that Carneal provides absolutely no teaching or suggestion for storing server pages within a cache located within a web server. Furthermore, Appellants have proved that Carneal cannot be modified or combined to provide such teaching. Therefore, Appellants assert that the proposed combination of Helgeson and Carneal is erroneous and cannot be relied upon to render the above-mentioned limitations of present claims 1, 9, 17 and 18 unpatentable.

3. **Carneal and Helgeson each fail to provide teaching, suggestion or motivation for a computer program product, web server or method for caching a parent and child server page, where the method includes storing only one copy of the code for the child server page in a cache located within a web server.**

Independent claim 9 recites in part, "[a] method for caching a parent and a child server page, comprising… storing code for the parent server page in a cache located internal to a web server… [and] storing only one copy of the code for the child server page in the cache." Independent claim 17 (a computer program product) and independent claim 18 (a web server) recite similar limitations. In this manner, the Specification and present claims provide a means for improving cache capacity by requiring that only <u>one</u> copy of the child server page be cached, regardless of the number of times it is called in the parent server page.

Statements in the final Office Action suggest that Carneal discloses a method for caching parent and child server pages, where the method includes "storing only one copy (e.g., col. 6, lines 36-44, an inline object is a child page) of the code for the child server page in the cache" (Final Office Action, page 5). The Appellants disagree, for at least the reasons set forth below.

Though Carneal states that prefetching "involves caching documents that are likely to be requested by a client" in the above-referenced passage (col. 6, lines 36-44), Carneal is silent as to the <u>number</u> of copies of child server page code that may be stored within the cache. One simply cannot assume that Carneal intends to store only <u>one</u> copy of a child server page, as presently claimed, when Carneal lacks explicit or implicit teaching, suggestion or even motivation for doing so.

As noted above, Helgeson is merely relied upon to show that caching can be performed on the web server side. Since Helgeson provides no other teachings relevant to the presently claimed case, the combination of Helgeson and Carneal cannot be relied upon to render the above-mentioned limitation of present claims 9, 17, and 18 unpatentable.

For the foregoing reasons, Appellant asserts that independent claims 1, 9, 17, and 18, as well as claims dependent therefrom, are patentably distinct over Carneal and Helgeson. Contrary to the characterizations made in the various Office Actions, the cited references cannot be properly combined or modified to provide teaching for all limitations recited in claims 1, 9, 17, and 18. Accordingly, Appellants assert that a *prima facie* case of obviousness has not been duly set out and, therefore, request that this rejection be reversed.

* * *

For the foregoing reasons, it is submitted that the Examiner's rejection of claims 1-3, 5-15, 17 and 18 was erroneous, and reversal of the decision is respectfully requested.

The Commissioner is authorized to charge the required fees to Daffer McDaniel LLP deposit account number 50-3268/5468-05300.

<div style="margin-left: 50%;">

Respectfully submitted,
/Kevin L. Daffer/
Kevin L. Daffer
Reg. No. 34,146
Attorney for Appellant

</div>

Daffer McDaniel, LLP
P.O. Box 684908
Austin, TX 78768-4908
Date: <u>August 23, 2006</u>
JMF

VIII.   CLAIMS APPENDIX

The present claims on appeal are as follows.

1.   A software system supporting distributed web applications, comprising:

a parent server page, containing a call to a child server page;

a cache within a web server, containing code for the parent server page and child server page, wherein the code for the parent server page does not contain all the code for the child server page; and

a link associated with the call to the child server page, and encapsulating information for locating and executing the code for the child server page.

2.   The software system as recited in claim 1, wherein the child server page may be executed using the link, without executing the parent server page.

3.   The software system as recited in claim 1, wherein the link further comprises a web page address and a list of request attributes.

5.   The software system as recited in claim 1, further comprising an instruction sequence that may be invoked to locate the child server page in the cache.

6.   The software system as recited in claim 1, further comprising an object-oriented software system.

7.   The software system as recited in claim 1, wherein a server page comprises a Java Server Page (JSP).

8.   The software system as recited in claim 7, wherein the child JSP may be executed in response to a request made to the web server by a client or another web server.

9.   A method for caching a parent and a child server page, comprising:

storing code for the parent server page in a cache located internal to a web server, such that the code for the parent server page does not contain all lines of code for the child server page;

storing only one copy of the code for the child server page in the cache;

creating in the code for the parent server page a link to the singular copy of the code for the child server page for locating and executing the code for the child server page; and

associating the link with more than one call to the child server page to execute from the cache a plurality of the singular copy of the code for the child server page.

10.   The method as recited in claim 9, wherein a server page comprises a Java Server Page (JSP).

11.   The method as recited in claim 10, further comprising invoking an instruction sequence to locate the code for the child JSP in the cache, in response to a request made by a web browser.

12.   The method as recited in claim 10, further comprising executing the code for the child JSP using the link, without executing all the code for the parent JSP.

13.   The method as recited in claim 12, wherein the child JSP is executed in the web server in response to a request made by a client or another web server.

14. The method as recited in claim 13, wherein the child JSP is executed only if it cannot first be located in the cache.

15. The method as recited in claim 13, wherein the cached child JSP may be updated without also updating the parent JSP.

17. A computer program product in a computer readable medium for use in storing a parent and a child server page in a cache, the computer program product comprising:

    instructions for storing code for the parent server page in the cache located within a web server, such that the code for the parent server page does not contain all lines of code for the child server page;

    instructions for storing only one copy of the code for the child server page in the cache;

    instructions for creating in the code for the parent server page a link to the singular copy of the code for the child server page for locating and executing the code for the child server page; and

    instructions for associating the link with more than one call to the child server page to execute from the cache a plurality of the singular copy of the code for the child server page.

18. A web server including memory and processor comprising;

    means for storing code for the parent server page in a cache located within the web server such that the code for the parent server page does not contain all lines of code for the child server page;

    means for storing only one copy of the code for the child server page in the cache;

means for creating in the code for the parent server page a link to the singular copy of the code for the child server page for locating and executing the code for the child server page; and

means for associating the link with more than one call to the child server page to execute from the cache a plurality of the singular copy of the code for the child server page.

IX. EVIDENCE APPENDIX

No evidence has been entered during the prosecution of the captioned case.

X. RELATED PROCEEDINGS APPENDIX

The related application listed below shares a common specification with the application currently on appeal, and is also under appeal at this time.

09/740,531   Appeal Brief filed March 21, 2005

However, because dissimilar art is cited in the present application and the above-mentioned related application, Appellants do not believe that the outcome of this appeal will have any bearing on the Board's decision on the related appeal. No other appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.